



Holistic CoDesign

Arun Rodrigues

**Scalable Computer Architecture Group
Sandia National Labs**

R&A: 5294117

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.



View of the Co-Design Problem

Scale.....

Many
Cores
+
Memory

X

Many
Many
Nodes

X

Many
Many
Many
Threads

Multiple Audiences.....

Network
Processor
System

X

Application writers
purchasers
designers

X

system procurement
algorithm co-design
architecture research
language research

X

present systems
future systems

Complexity.....

Multi-Physics Apps
Informatics Apps

X

Communication Libraries
Run-Times
OS Effects

X

Existing Languages
New Languages

Constraints.....

Performance
Cost

Power
Reliability



Cooling
Usability

Risk
Size

Hidden Worldwide Impact

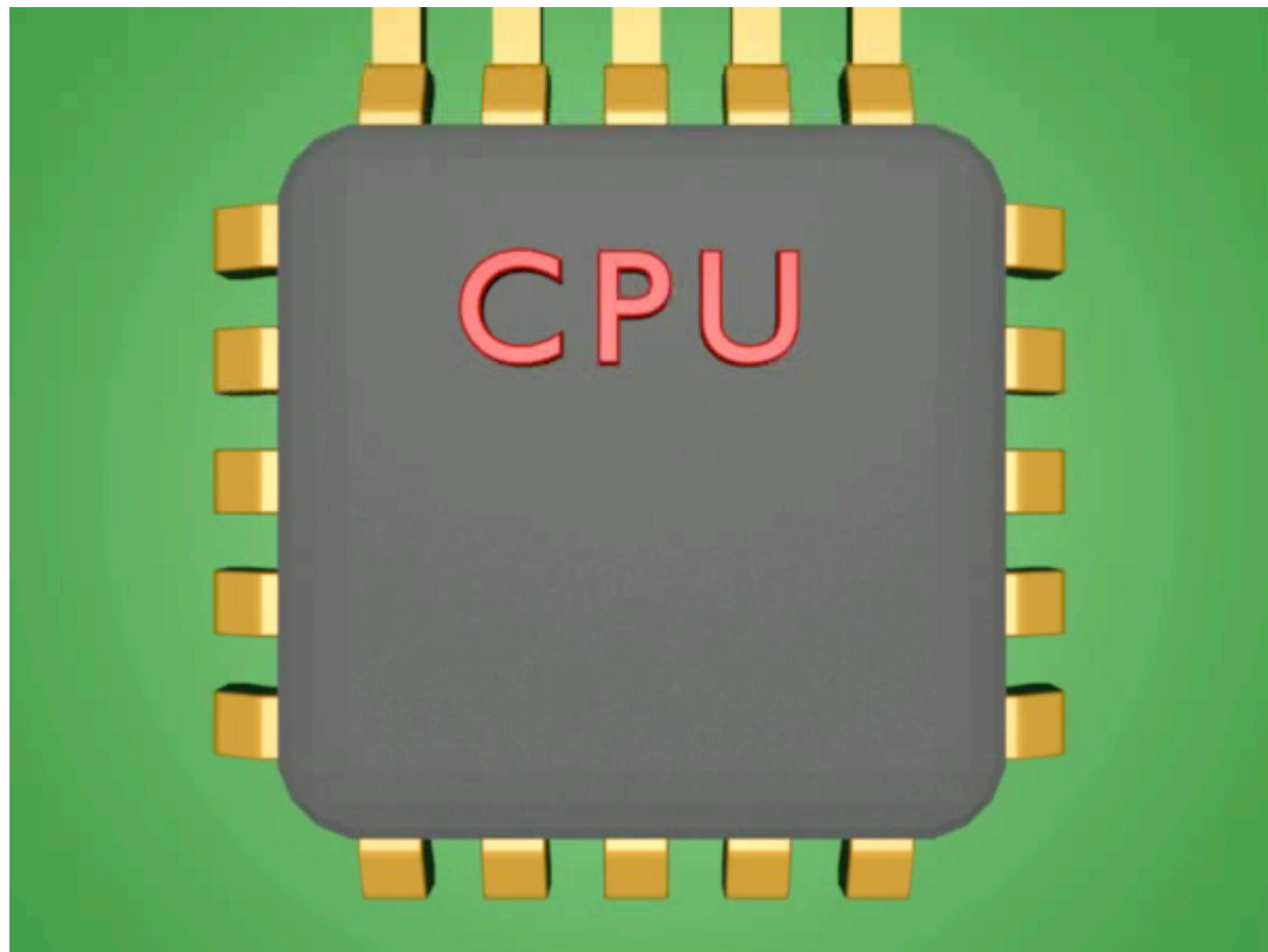
**"Total power used by servers [in 2005] represented ...
an amount comparable to that for color televisions. "**

**-ESTIMATING TOTAL POWER CONSUMPTION BY SERVERS IN THE U.S. AND
THE WORLD, Jonathan G. Koomey**

3741e9 KW-Hrs	Total US power consumption
* 3-4%	used by computers (>2% servers, >1% household computer use)
= 112 - 150e9 KW-Hrs	US Computer power consumption
* \$0.1 \$/KW-Hr	Retail cost, US Average 2009
= \$11 - \$15	Billion US\$ in compute power
* 3-5	in 2005 US was roughly 1/3 of servers, by power. This has probably decreased
= \$33 - \$75	Billion US\$ in worldwide computer power
=  - 	Yearly GDP of Qatar to Burma

Case Study: Memory is efficient Memory Access is Inefficient

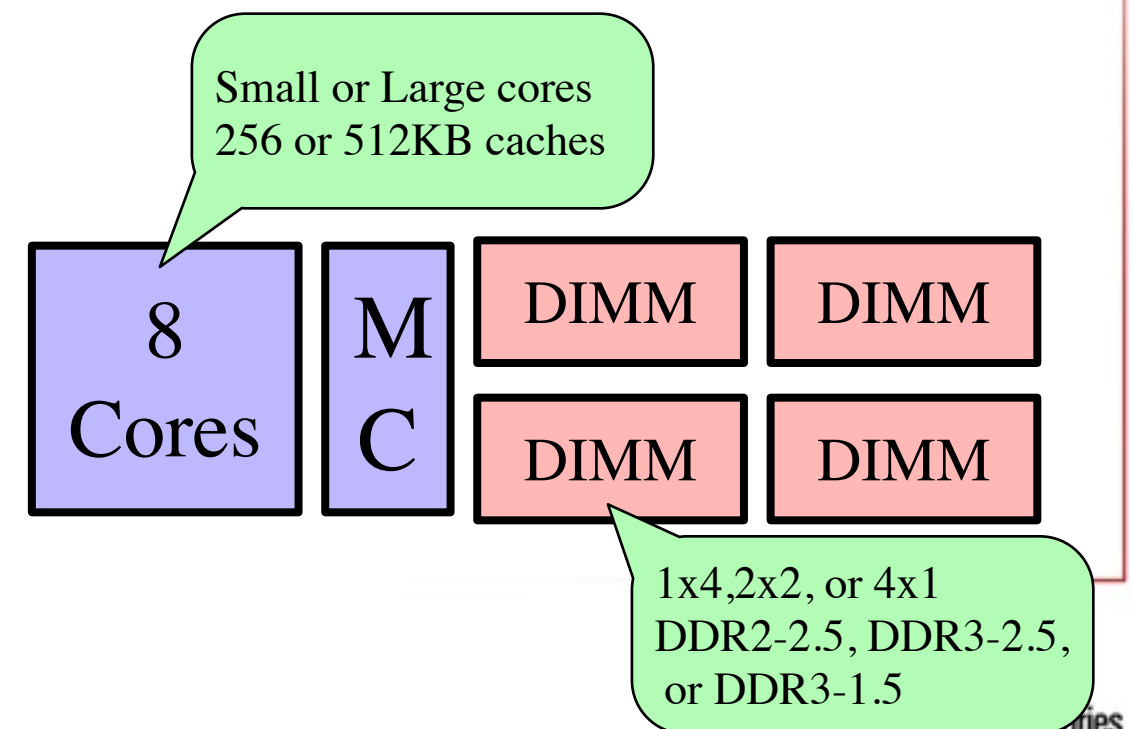
- DRAM cells require < 1 pJ to access
- Current DRAM architectures are not power efficient
- Long distances \rightarrow high power
- We pay for more than we get at every level
 - Cache: throw away 75-80%
 - DRAM Row: Charge 1024B for each 64B access
 - DIMM: Charge 8-9 chips/access
 - ~ 800 pJ/byte total
- DRAM design driven by packaging constraints
 - $\sim 50\%$ of DRAM chip cost is packaging, mainly in pins
 - DIMMs use multiple chips with a few data pins to achieve high BW



Design Space Exploration Example

- **Design Space for Multiple applications machine**
 - **Inputs: Memory Channels, Memory Technology, Core type, Cache size (144 configurations)**
 - **Outputs: Energy, Performance, Cost**
- **Methodology**
 - **Performance models: genericProc, DRAMSim2**
 - **Energy Models: DRAMSim2, McPAT**
 - **Cost Models: IC Knowledge**
 - **Find Pareto-optimal set for Energy, Performance, and Power for each application**

Parameter	Large Core	Small Core
Instruction Fetch/Decode Width	4	2
Instruction Issue Width	8	4
Instruction Issue	Out of Order	In Order
Instrucion Commit Width	8	4
FPU's	2	1
Maximum Instructions 'In-Flight'	128	64
Load Store Queue	64	32
Energy/Instruction (pJ)	3365	1245
Si Area (8 cores, no cache)		
Si Cost (8 cores, Large cache)	\$78.79	\$68.53
Si Cost (8 cores, Small cache)	\$57.05	\$49.51



Design Space Exploration Results

- Latest memory technology not always best (DDR2 beats DDR3) due to latency, cost
- For these apps & inputs, fewer memory channels is better
- No “best” processor - depends on tradeoff between cost, performance, energy
- Better understanding of which configurations are best for a given application
- Can be used as basis for application optimization

Chan.	Memory	Core	Cache	Energy	Performance	Cost
1	DDR2 25	Large	Large	1.00	1.000	206.14
1	DDR2 25	Small	Small	1.00	0.464	176.86
1	DDR2 25	Small	Large	1.03	0.532	195.88
1	DDR2 25	Large	Small	1.49	0.902	184.40

Pareto Optimal Designs

Design Space Exploration Results

- Latest memory technology not always best (DDR2 beats DDR3) due to latency, cost
- For these apps & inputs, fewer memory channels is better
- Better understanding of which configurations are best for a given application

Application	Chan.	Memory	Core	Cache	Energy	Performance	Cost
HPCCG	1	DDR2 25	Small	Small	250	510.7	176.86
HPCCG	1	DDR2 25	Small	Large	253	541.6	195.88
HPCCG	1	DDR3 25	Small	Large	263	566.9	220.20
HPCCG	1	DDR3 15	Small	Large	318	585.4	241.48
MD	1	DDR2 25	Large	Large	1504	105.9	206.14
MD	1	DDR2 25	Small	Small	1106	49.7	176.86
MD	1	DDR2 25	Small	Large	1119	50.7	195.88
MD	1	DDR2 25	Large	Small	1579	102.0	184.40
MD	2	DDR2 25	Large	Large	1480	105.4	213.55
MD	2	DDR2 25	Small	Small	1079	49.6	184.27
MD	2	DDR2 25	Small	Large	1093	50.6	203.29
gups	1	DDR2 25	Large	Small	1777	7.2	184.40
gups	1	DDR2 25	Small	Small	1183	6.9	176.86
gups	2	DDR2 25	Small	Small	1114	6.6	184.27
pagerank	1	DDR2 25	Large	Large	751	162.4	206.14
pagerank	1	DDR2 25	Small	Small	667	49.4	176.86
pagerank	1	DDR2 25	Small	Large	565	64.1	195.88
pagerank	1	DDR2 25	Large	Small	867	126.2	184.40
pagerank	2	DDR2 25	Large	Large	748	151.0	213.55

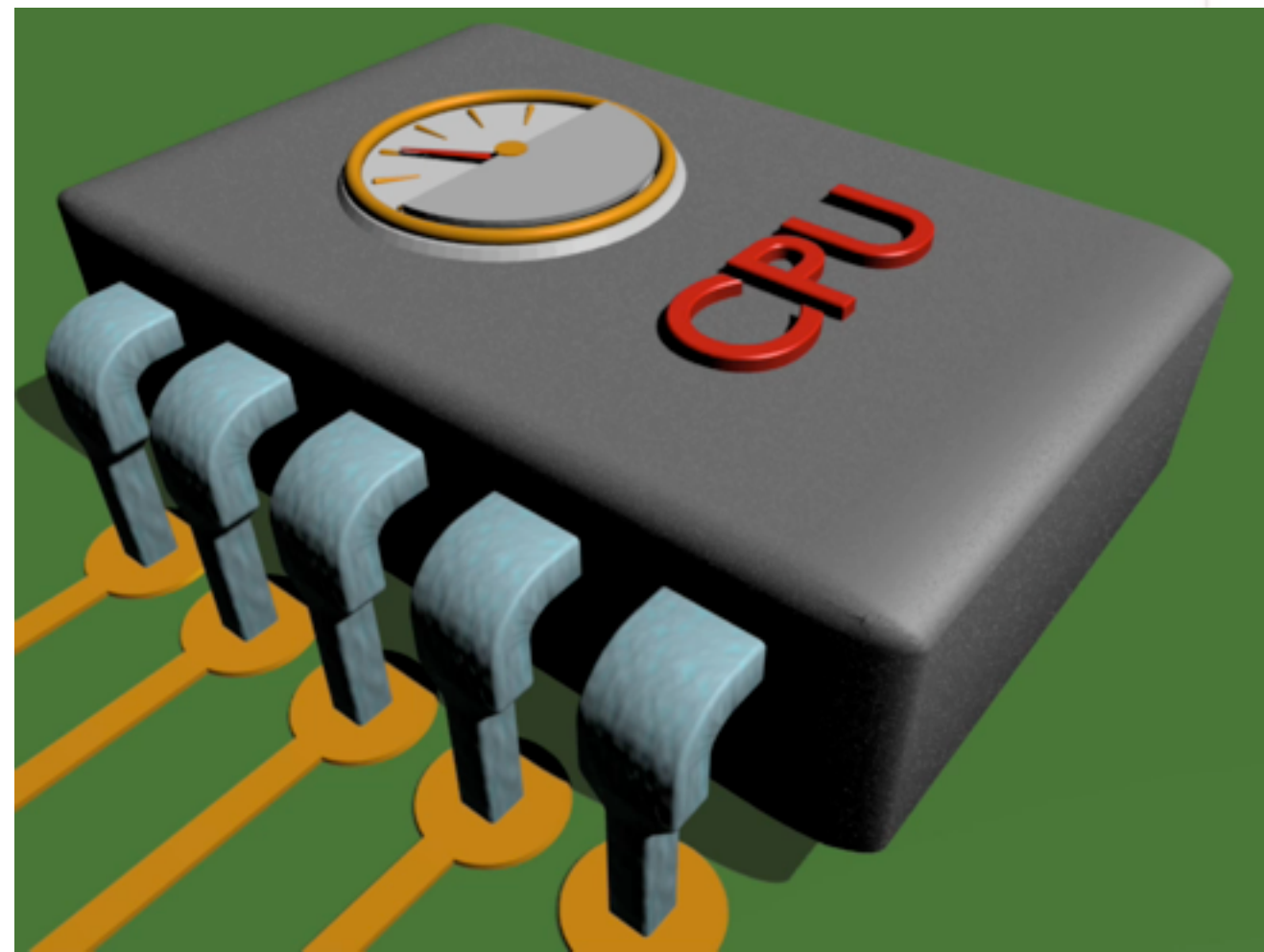
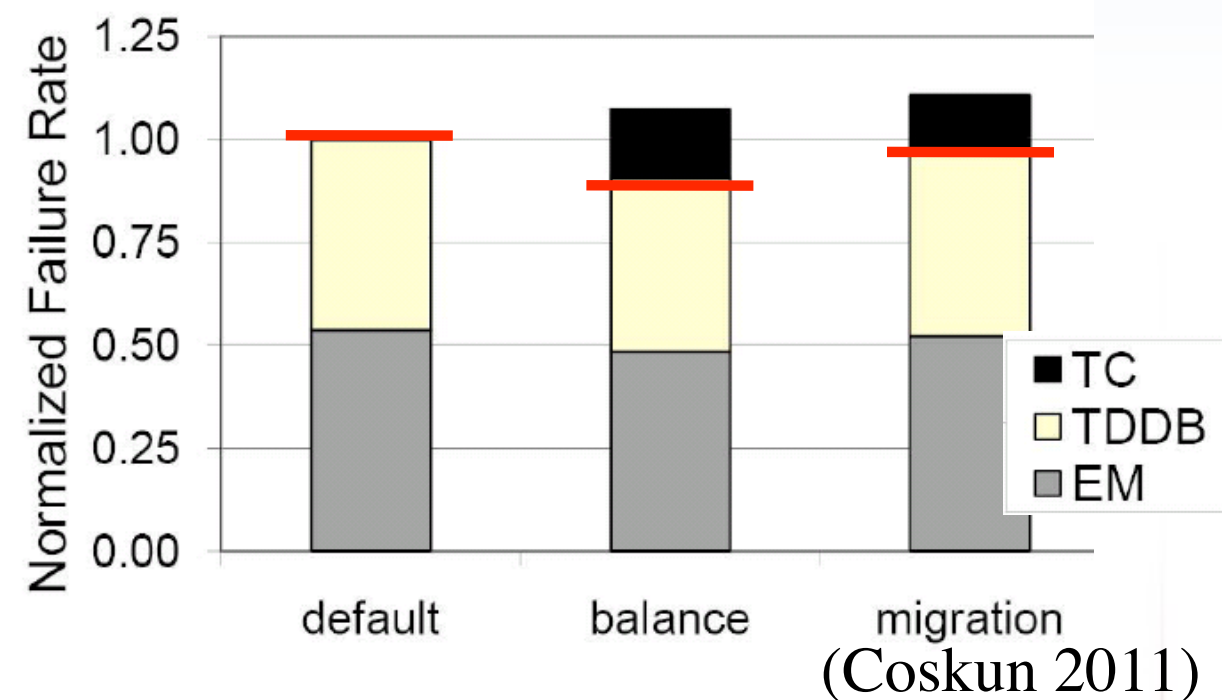
Pareto Optimal Designs

Case Study: Reliability vs. Power

Hidden cost of DVFS

- **Dynamic voltage/frequency Scaling reduces power**
- **→ Reduces temperature**
- **→ Causes thermal cycling**
- **→ Reduces reliability**

- **Need**
 - Algorithms to balance temperature, lower power, & maintain performance
 - Arch: Sensors and feedback
 - Runtime: Scheduler changes
 - App: Awareness



Case Study: Scratchpads vs. Caches

- **Power**

- 32KB 45nm 4-way cache: 142 pJ/read
- 32KB 45nm SRAM: 24pJ/read

- **Performance**

- Scratchpads: Predictable, interface well with DMA
- Caches: Better average performance, requires less application knowledge

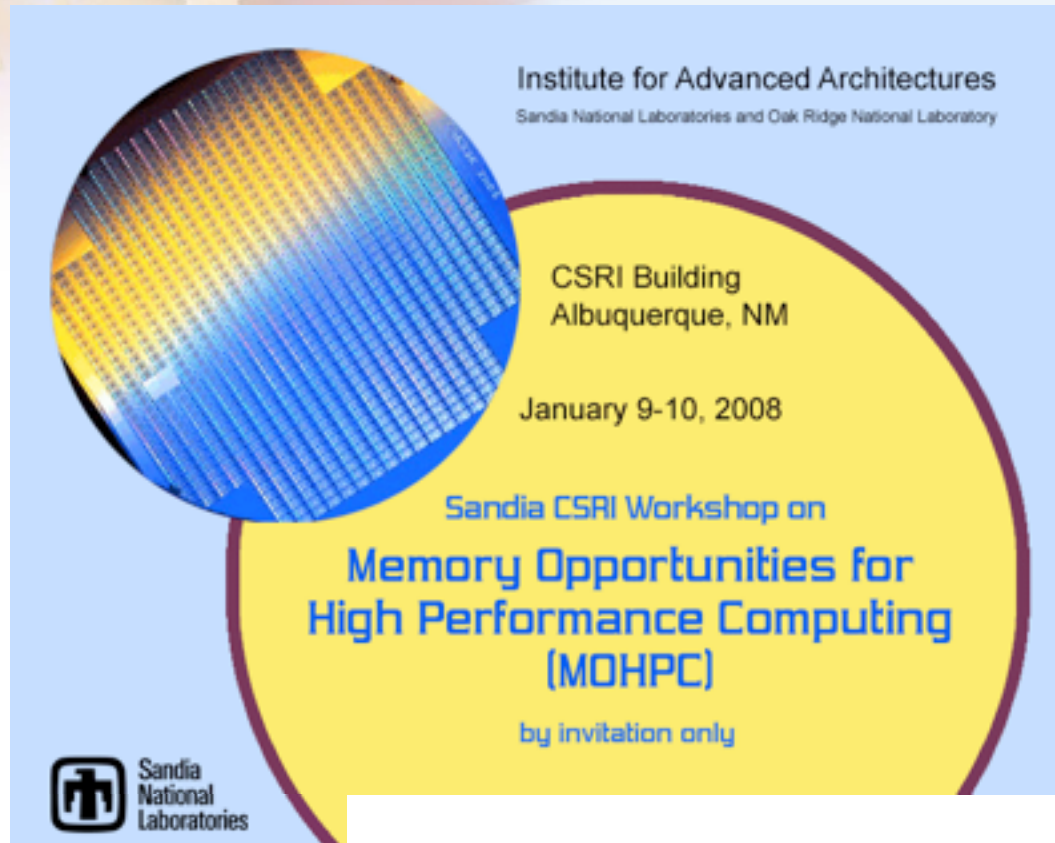
- **Programmability**

- Scratchpads are usable, as demonstrated by the CELL and embedded community
- Scratchpads are difficult to use, as demonstrated by the CELL and embedded community

- **Turf Wars**



Turf Wars



Arch

App

Runtime

- **January 2008 MOHPC Workshop**
- **Look at memory opportunities for future HPC systems**
- **Three groups**
 - **Architecture**
 - **Runtime (libraries, runtime, & OS)**
 - **Applications**
- **Each came up with recommendations**
 - **Each group independently brought up scratchpads**
- **Exchange & Critique**
 - **Generate positive consensus**

Turf Wars: Architects View

- **Scratchpad:** While feasible, a key concern is that saving state is difficult and expensive. Additionally, a scratchpad presents resource contention issues. A lockable cache may be easier to implement and manage. A key question for the application writers is to express why they want a scratch pad. Is it because it makes naming easier? Is it for bandwidth? latency? guaranteed timing? Also, there are the standard concerns about portability.

Translation

- **We can make a scratchpad, no problem**
- **But, the apps people don't really know how to use one**
- **And, the runtime people won't let them have it anyway**



Turf Wars: Runtime View

The use of “local” memory (scratch pads, etc.) shows significant promise, but tends to also be performed in a non-portable fashion. Additionally, there tend to be few mechanisms for coping with the expansion of the memory hierarchy.

Translation

- **We can deal with a scratchpad**
- **But, the apps people don't like 'em**
- **And, the architects won't give one to us anyway**



Turf Wars: Application View

highly desirable, however, historically it has proven very difficult to generate a robust, portable, non-ephemeral API to support these features.

Translation

- We love scratchpads
- But, the runtime people won't let us access them
- And, the architects won't give one to us anyway





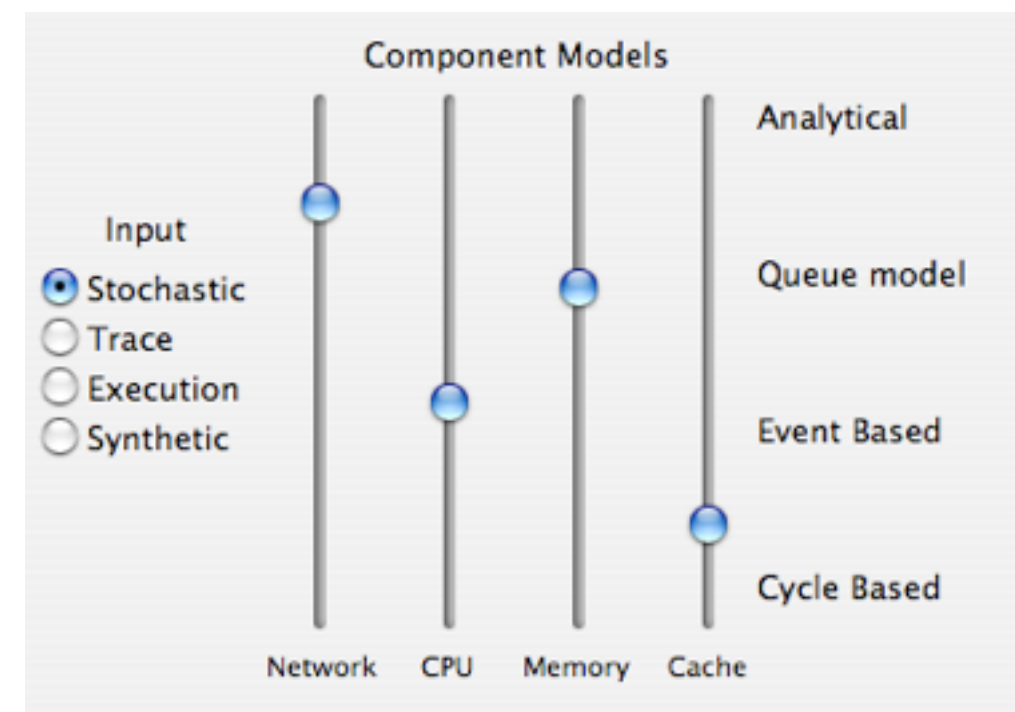
What is needed

- **Simulation/Emulation environments**
 - Parallel, Scalable, & Multi-scale
 - Holistic
 - Open, Trusted, and Accepted
- **Methodologies**
 - Validation
 - Multi-scale mix-n-match

Multi-Scale

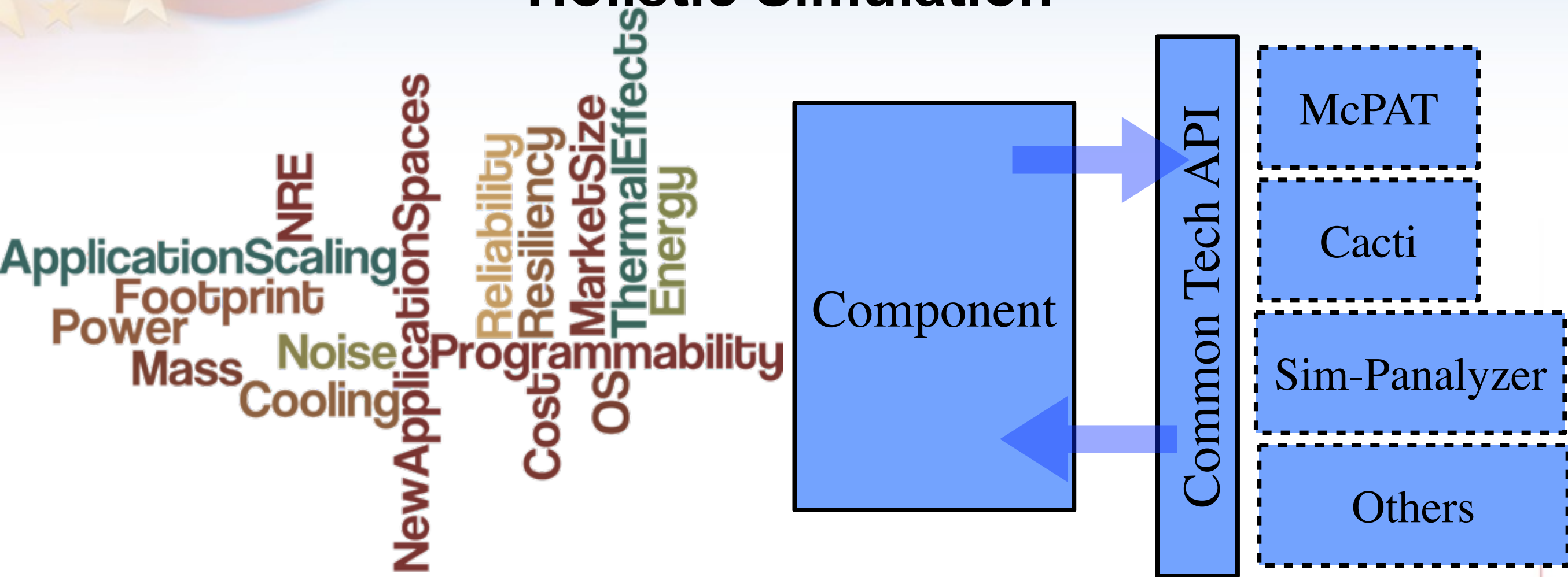
- **Goal: Enable tradeoffs between accuracy, flexibility, and simulation speed**
 - No single “right” way to simulate
 - Support multiple audiences
- **High- & Low-level interfaces**
 - Allows multiple input types
 - Allows multiple input sources
 - Traces, stochastic, state-machines, execution...

	High-Level	Low-Level
Detail	Message	Instruction
Fundamental Objects	Message, Compute block, Process	Instruction, Thread
Static Generation	MPI Traces, MA Traces	Instruction Trace
Dynamic Generation	State Machine	Execution



Multiscale Parameters

Holistic Simulation



- Design space includes much more than simple performance
- Create common interface to multiple technology libraries
 - Power/Energy
 - Area/Timing estimation
- Make it easier for components to model technology parameters

SST Simulation Project Overview

Goals

- Become the standard architectural simulation framework for HPC
- Be able to evaluate future systems on DOE workloads
- Use supercomputers to design supercomputers

Status

- Current Release (2.1) at code.google.com/p/sst-simulator/
- Includes parallel simulation core, configuration, power models, basic network and processor models, and interface to detailed memory model

Technical Approach

- Parallel
 - Parallel Discrete Event core with conservative optimization over MPI
- Holistic
 - Integrated Tech. Models for power
 - McPAT, Sim-Panalyzer
- Multiscale
 - Detailed and simple models for processor, network, and memory
- Open
 - Open Core, non viral, modular

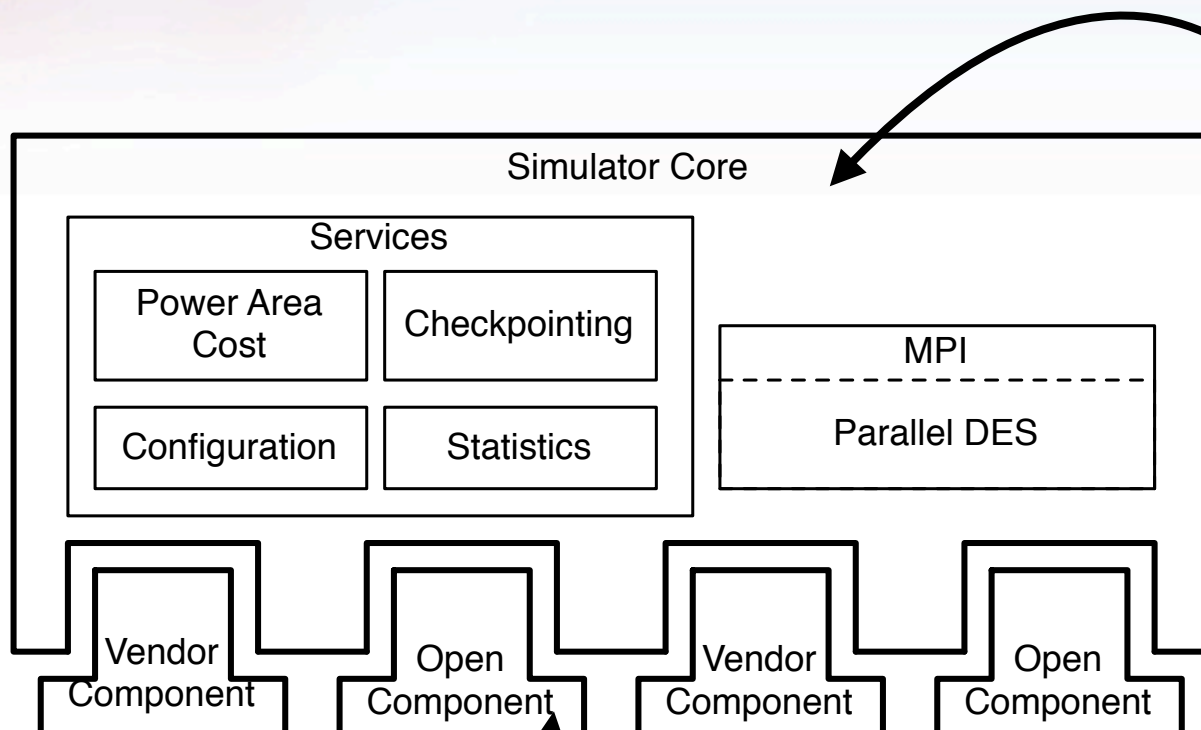


Consortium

- “Best of Breed” simulation suite
- Combine Lab, academic, & industry



Open Simulator Framework



• Simulator Core will provide...

- Power, Area, Cost modeling
- Checkpointing
- Configuration
- Parallel Component-Based Discrete Event Simulation

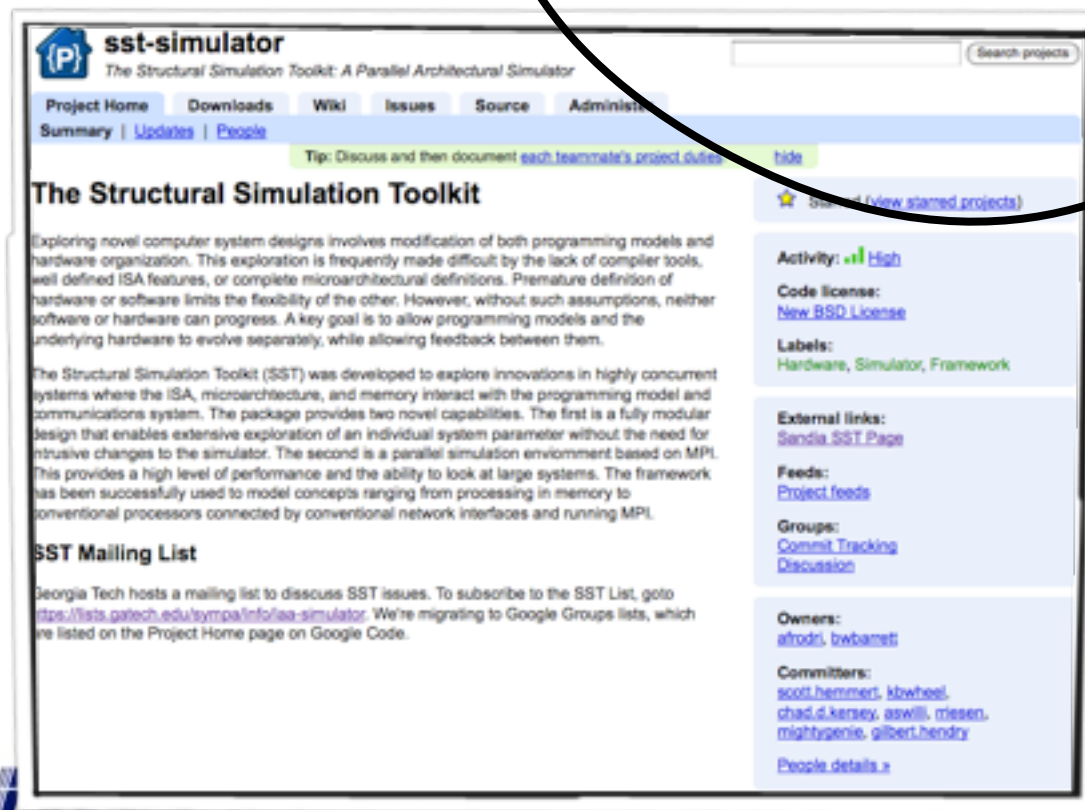
• Components

- Ships with basic set of open components
- Industry can plug in their own models

- Under no obligation to share

• Open Source (BSD-like) license

• SVN hosted on Google Code





What is needed

- **Validation Methodologies**

- Where are our error bars?
- How much error can we live with?
- What is the standard for validation of things which do not exist?

- **Multi-scale Methodologies**

- Can we mix and match simulation models of different scale?
- When we mix a high-fidelity and low-fidelity model how is the error effected?

The New Project Polygon



- “Fast Cheap or Good” no longer enough

- New Factors

- Resilience
- Risk
- Programmability
- Power
- Energy
- Cost (purchase vs. TCO)
- Commercial adoption
- “Social” Issues

- Community needs tools!

- Simulation
- methodologies

